

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning documents *will not* correct images,  
Please do not report the images to the  
Image Problem Mailbox.

This Page Blank (uspto)

(19)



Europäisches Patentamt

European Patent Office

Offic Européen des brevets

(11)



EP 0 887 981 A2

(12)

## EUROPEAN PATENT APPLICATION

(43) Date of publication:

30.12.1998 Bulletin 1998/53

(51) Int Cl. 6: H04L 29/06

(21) Application number: 98304869.5

(22) Date of filing: 19.06.1998

(84) Designated Contracting States:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE

Designated Extension States:

AL LT LV MK RO SI

(30) Priority: 26.06.1997 US 883636

(71) Applicant: SUN MICROSYSTEMS, INC.  
Mountain View, California 94043-1100 (US)

(72) Inventor: Gong, Li

Menlo Park, California 94025 (US)

(74) Representative: Read, Matthew Charles et al

Venner Shipley & Co.

20 Little Britain

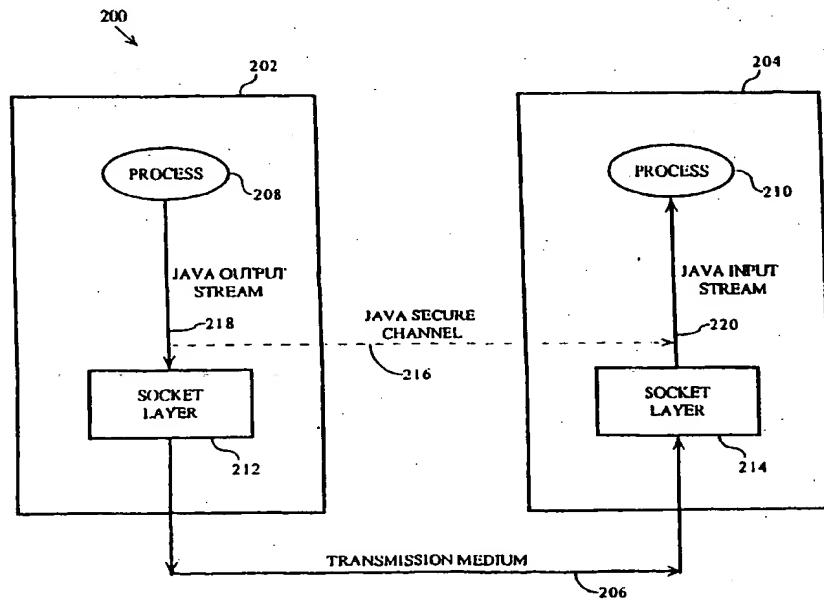
London EC1A 7DH (GB)

### (54) Layer-independent security for communication channels

(57) A method and apparatus for providing layer-independent secure network communication is provided. According to an embodiment of the invention, a transmission medium is provided between a first network node and a second network node. Both the first network node and the second network node support at least one common communication protocol. A Java output stream is established between a first process executing on the

first network node and the transmission medium. Also, a Java input stream is established between a second process executing on the second multilayered node and the transmission medium. Data to be transmitted from the first process to the second process is encrypted by the first process and written to the Java output stream. The data is transmitted to the second network node. Then the data is read from the Java input stream by the second process and decrypted.

FIG. 2



EP 0 887 981 A2

## D description

### FIELD OF THE INVENTION

The invention relates to data security, and more specifically, to a method and apparatus for providing layer-independent security in network communications.

### BACKGROUND OF THE INVENTION

Some communication networks, particularly complex ones, support multiple communication protocols or "layers." Each layer specifies some functionality or "service" of the network and interacts with the layers immediately above and below, using services of the layer immediately below, while providing services to the layer immediately above. The lowest layer in a communication network typically governs direct communication between the hardware at different network nodes, while the highest layer handles direct communication with application programs executing on the network nodes.

The layered approach to implementing communication networks simplifies the creation and modification of complex communication architectures by providing for incremental changes on a layer-by-layer basis, which are transparent to other layers in the architecture. Two examples of layered communication protocols are the Transmission Control Protocol/Internet Protocol (TCP/IP), which has five layers, and the International Standards Organization's (ISO) Open Systems Interconnection (OSI) Reference Model (RM), which has seven layers.

The proliferation of communication networks and increased frequency of security breaches has underscored the importance of providing secure network communications. Many communication networks depend upon a secure communication connection or "channel" to maintain security. In the context of secure communication networks, a secure communication channel is a connection which provides for the encryption, authentication or otherwise secure transmission of data between network nodes.

Sometimes, setup negotiation is used to establish security for a communication channel. In the context of network communications, setup negotiation refers to specifying and agreeing to the details about security for a communication channel, such as the details of a particular encryption scheme to be used. Once setup negotiation is complete, all communication during the session conforms to the agreed upon security protocol, which provides secure communication.

Setup negotiation is an effective tool for providing secure communication during a communication session. However, when the amount of information included in each session is small, for example when a session contains only a single message, then the overhead attributable to setup negotiation can adversely affect communication performance. Moreover, some communica-

tion architectures do not include a session layer, which requires that a session layer be added to support session type security, further degrading performance.

Another approach for providing a secure communication channel involves encrypting or encoding data at a specific layer on a transmitting network node and then decrypting or decoding the data at a corresponding layer on a destination network node. Encrypting data at a specific layer typically involves applying an encryption algorithm based upon the format of data at a particular layer. Header data added by higher layers is also encrypted. Layer-specific encryption is particularly useful in datagram-based or packet-based networks which are typically sessionless and encapsulate data in datagram packets or some other type of data packet. For example, header data may be added to a data packet so that the data packet conforms to a particular format. This approach also provides for multiple encryptions to be performed at different layers.

Although layer-specific encryption can provide a secure communication channel while avoiding the overhead penalty associated with setup negotiation, it does have several limitations. First, all encryption and decryption must occur at the same corresponding layer on both the transmitting and receiving network nodes, according to the specific protocol supported by that layer. For example, Simple Key Management for Internet Protocols (SKIP) is designed to be used with internet protocol packets at the network layer, which requires internet layer specific function calls. On the other hand, Netscape Communications Corporation's Secure Sockets Layer (SSL) is designed to be used at the (Unix) socket layer and requires socket layer-specific function calls to encrypt and decrypt data. The result is that one application implementing security according to SKIP cannot interact with another application implementing security according to SSL.

In addition, layer-specific encryption can be difficult to employ in object-oriented environments because of the inherent level of abstraction required. For example, some layers operate on data bytes, which often is a much lower level than objects in an object oriented environment.

In view of both the need to provide secure communication channels and the limitations in the prior approaches, an approach for providing a secure communication channel which does not rely upon layer-specific encryption and which does not require setup negotiation is highly desirable.

### SUMMARY OF THE INVENTION

According to one aspect of the invention, a method provides communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node. Both the first network node and the second network node each support at

least one common communication protocol. According to the method, a communication channel is established between the first network node and the second network node. Then, a first stream is established between the first process and the communication channel.

In the context of the invention, a "stream" is an abstraction which refers to the transfer or "flow" of data, in any format, from a single source, to a single destination. A stream typically flows through a channel or connection between the sender and receiver, in contrast to data packets, which are typically individually addressed and which may be routed independently to multiple recipients. Hence, an application can write data to, or read data from, a stream without knowing the actual destination or source, respectively, of the data.

After the first stream is established between the first process and the communication channel, a second stream is established between the second process and the communication channel. Data to be transmitted between the first and second processes is encrypted. The encryption of the data is independent of the communication protocol supported by the first network node. The encrypted data is then written to the first stream which causes the encrypted data to be transmitted from the first network node to the second network node. The encrypted data is read from the second stream and then decrypted to obtain decrypted data which is identical to the data on the first network node before the data was encrypted.

#### BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

Figure 1 is a block diagram of a multi-layered communication network according to an embodiment of the invention;

Figure 2 is a block diagram of a multi-layered communication network according to another embodiment of the invention;

Figure 3 illustrates a stream format according to an embodiment of the invention;

Figure 4 is a flow chart illustrating a method for providing layer-independent secure communication in a multi-layered communication network according to an embodiment of the invention;

Figure 5 is a block diagram of a Java secure channel arrangement according to an embodiment of the invention; and

Figure 6 is a block diagram of a computer system on which the invention may be implemented.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A method and apparatus for providing layer-independent secure communications in a multi-layered communication network is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, the invention may be practiced without these specific details. In other instances, well-known structures and devices are illustrated in block diagram form in order to avoid unnecessarily obscuring the invention.

#### 15 FUNCTIONAL OVERVIEW

The invention provides a method and apparatus for providing layer-independent secure communications in a multi-layered communication network. In general, a communication channel or connection is first established between a first multi-layered network node and a second multi-layered network node. Then, a first stream is established between a first process, executing on the first multi-layered network node, and the communication channel. A second stream is then established between a second process, executing on the second multi-layered network node and the communication channel. Then, the first process performs a layer-independent encryption of data to be transmitted between the first and second multi-layered network nodes and then writes the encrypted data to the first stream, which causes the encrypted data to be transmitted to the second multi-layered network node. Then, the encrypted data is read by the second process from the second stream and decrypted so that the decrypted data is identical to the data on the first multi-layered network node prior to being encrypted.

Figure 1 illustrates a multi-layered communication network 100 to which the invention is applicable. In general, multi-layered communication network 100 includes multi-layered nodes 102, 104, communicatively coupled by transmission medium 106. Although multi-layered communication network 100 may resemble the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model (RM), the invention is applicable to any multi-layered communication network.

A process 108 executes on multi-layered node 102 while a process 110 executes on multi-layered node 104. Multi-layered node 102 supports a multi-layered communication hierarchy 112, where each identified layer supports a particular communication protocol. Each layer in hierarchy 112 offers certain services to the higher layers while shielding the higher layers from the details of how those services are actually implemented. Multi-layered node 104 also supports a multi-layered communication hierarchy 114, which includes layer corresponding to the layers in hierarchy 112. All data trans-

mitted from process 108 to transmission medium 106 conforms to all communication protocols supported by hierarchy 112.

For example, to transmit data 116 from process 108 to transmission medium 106, data 116 must first conform to an application protocol specified by application layer 118 on multi-layered node 102. According to one embodiment of the invention, this requires that data 116 be formatted according to application layer 118 protocol and that an application protocol header AH be appended to the front end of data 116 which specifies the format of data 116.

This process is repeated for each layer in hierarchy 112. According to one embodiment of the invention, the formatting of data 116 according to a data link layer 120 involves the addition of both a header portion DH and a trailer portion DT to a data portion 122. It should be noted that data link layer 120 is not aware of which portion of data portion 122 corresponds to data 116 and which portion represents formatting information of higher layers. Data link layer 120 formats the entire data portion 122 without regard to which portion may be "real" data 116 and which portion is formatting information added by higher layers in hierarchy 112.

When messages are received by multi-layered node 104 from transmission medium 106, a reverse process occurs. Since messages must conform to application layer protocol before being processed by process 110, any formatting information attributable to layers below application layer 128 must be removed.

As previously discussed, one approach for providing secure communication between process 108 and process 110 is to have processes 108, 110 perform set-up negotiation prior to transmitting data. However, this approach can adversely affect data throughput, particularly when the setup negotiation is performed on a packet-by-packet basis.

Another previously discussed approach is to encrypt the data at one of the layers in hierarchy 112 on multi-layered node 102 before transmitting the data on transmission medium 106. Then, after the encrypted data is received on node 104, the data is decrypted at the corresponding layer in hierarchy 114 on multi-layered node 104 before the data is received by process 110. For example, data may be encrypted at the network layer 124 on multi-layered node 102 and then decrypted at network layer 126 on multi-layered node 104 on a packet-by-packet basis. Although this approach is robust from a security standpoint, the data must be decrypted at the same layer at which the data was encrypted.

#### LAYER-INDEPENDENT SECURITY

An approach which provides layer-independent secure network communication in a multi-layered communication network according to an embodiment of the invention is illustrated by the block diagram of Figure 2. A multi-layered communication network 200 includes mul-

ti-layered nodes 202, 204 which are communicatively coupled by a transmission medium 206. A process 208 executes on multi-layered node 202 while a process 210 executes on multi-layered node 204.

5 Multi-layered nodes 202, 204 each support one or more communication layers (protocols) including socket layers 212, 214, respectively. Socket layers 212, 214 provide an interface between processes 208, 210, respectively, and transmission medium 206. Multi-layered 10 nodes 202, 204 may support addition layers (not illustrated) both above and below socket layers 212, 214. Accordingly, socket layers 212, 214 each include sockets (not illustrated), which are end points similar to an OSI Transport Service Access Point (TSAP), and which 15 provide a connection between layers above and below socket layers 212, 214. In addition, a Java secure channel 216 is provided between node 202 and node 204. Java security channel 216 provides for the layer-independent encryption of high level data constructs such 20 as objects.

Generally, according to an embodiment of the invention, layer-independent security for communications between process 208 and process 210 is provided by process 208 encrypting data which is then written to a 25 Java output stream 218. A Java stream is a stream which provides for the transfer of low level data constructs, such as bytes, as well as high level data constructs, such as serialized objects, between a source and a destination. The data is then conformed to a socket 30 layer protocol by socket layer 212 and written to transmission medium 206. The data is then processed according to socket layer protocol by socket layer 214 and read from a Java input stream 220 by process 210 and finally decrypted by process 210.

35 Encryption of stream data according to embodiments of the invention is by definition layer-independent and provides a level of abstractness which is compatible with many abstract processes and languages which support streams, such as object oriented languages. 40 Besides the layer-independent data encryption performed by process 208, additional (layer-dependent) encryption may be provided at any layer in node 202, with decryption being performed at the corresponding peer layer in node 204.

45 The data format of object output stream 218 and object input stream 220 is illustrated in Figure 3. Generally, stream format 300 is an abstract message format which is self-contained and layer-independent. Stream format 300 includes 1 to N variable length messages 50 (M1, M2... Mn). Each message (M1, M2... Mn) includes a header portion (H1, H2... Hn) and a data portion (DATA1, DATA2... DATA<sub>n</sub>). According to one embodiment of the invention, each header portion (H1, H2... Hn) specifies the length of the associated data portion 55 (D1, D2... Dn) and also includes encryption key/authentification information which eliminates the need for setup negotiation. However, certain encryption key/authentification information is established once during system

setup so that recipients of the messages ( $M_1, M_2 \dots M_n$ ) can decrypt data contained in the data portion ( $D_1, D_2 \dots D_n$ ) of each message ( $M_1, M_2 \dots M_n$ ).

The flexibility of stream format 300 of the invention provides for the implementation of various encryption/authentication approaches and is not limited to the particular encryption/authentication approach described herein. In addition, since stream format 300 is layer independent, various data formats may be employed without departing from the scope of the invention.

The specific steps for providing layer-independent security of network communication according to an embodiment of the invention are now described with reference to both the block diagram of Figure 2 and the flow chart of Figure 4. Generally, the steps are described in the context of an object oriented programming method associated with an object, contained in process 208, which invokes a method associated with a remotely located object contained in process 210. In the non-object oriented context, this is very similar to process 208 issuing a remote procedure call (RPC) to invoke a process remotely located on multi-layered node 204. For purposes of explanation, the data transmitted by the method associated with the object contained in process 208 which invokes the method associated with the remotely located object contained in process 210 is hereinafter referred to as the "object data."

After starting in step 400, in step 402, multi-layered nodes 202, 204 establish an encryption/authentication approach during system setup. Unlike traditional setup negotiation which must be continuously re-negotiated, such as on a per session basis, the agreed upon encryption/authentication approach established between multi-layered nodes 202, 204 only needs to be set up once during system setup, or when either multi-layered node 202, 204 is connected to another node and the security techniques described herein are to be employed with that other node.

In step 404, a Java secure channel 216 is established between node 202 and node 204. According to one embodiment of the invention, Java secure channel 216 is an object class which is defined and invoked by process 208.

In step 406, object output stream 218 is established between process 208 and socket layer 212, and in step 408, object input stream 220 is established between socket layer 214 and process 210. According to one embodiment of the invention, object output stream 218 is an object class defined by process 208 while object input stream 220 is an object class defined by process 210.

In step 410, the object data to be transmitted from process 208 to process 210 is serialized, sometimes referred to as "flattening the object," and then encrypted in step 412 based upon the encryption/authentication approach established in step 402.

In step 414, the object data (serialized and encrypted) is written to object output stream 218, which is re-

ceived by socket layer 212 and formatted according to socket layer protocol. In step 416, the object data is transmitted from socket layer 212 of multilayered node 202 to socket layer 214 of multi-layered node 204 over transmission medium 206.

As previously discussed, multi-layered node 202 is illustrated as having a single layer, socket layer 212, while multi-layered node 204 is illustrated as having a single layer, socket layer 214, for purposes of explanation. However, multi-layered nodes 202, 204 may be multi-layered and contain other layers above and below socket layers 212, 214. Consequently, although according to an embodiment of the invention, the object data is transmitted onto transmission medium 206 in the format illustrated in Figure 3, it is understood that additional formatting of the object data may be performed according to various other communication protocols contained in multi-layered nodes 202, 204. For example, if multi-layered node 202 also supports Internet protocol (IP), then each message (M1, M2...Mn) illustrated in Figure 3 would also contain IP header information.

25 After the object data is received by socket layer 214, the object data is read from object input stream 220 by process 210 in step 418. In step 420, the object data is decrypted according to the encryption/authentication approach established in step 402. Then, in step 422, the object data is de-serialized and the method associated with the object remotely located in process 210 is executed. Finally, the process is complete in step 424.

30 Although embodiments of the invention have been described in the context of encrypting and decrypting object data by processes 208, 210, which are effectively above all of the layers supported by multi-layered nodes 202, 204, respectively, data may be encrypted and de-  
35 crypted at any layer supported by multi-layered nodes 202, 204, since the encryption of data is performed before the data is written to a stream and is therefore layer-independent.

For example, referring again to Figure 1, according to another embodiment of the invention, process 108 encrypts data 116 and then writes data 116 to a stream (not illustrated) which is formatted according to the protocol hierarchy 112 and transmitted to multi-layered node 104 on transmission medium 106. Since data 116 was encrypted at the stream level, data 116 may be decrypted at any layer in hierarchy 114, so long as data 116 can be extracted from the data stream. Typically, the size and position of data 116 within a data chunk is known which allows data 116 to be extracted from a data chunk even though the data chunk contains protocol specific information from higher layers. However, if data 116 is encrypted at any other layer in hierarchy 112, then data 116 must first be decrypted at a corresponding layer in hierarchy 114.

55 According to another embodiment of the invention, a stream is connected to several other protocol-specific streams to support the broadcasting or multi-casting of encrypted information. Figure 5 illustrates an arrange-

ment 500 which includes a stream 502 according to an embodiment of the invention, connected via connectors 504, to intelligent converters 506, which convert stream 502 into protocol-specific streams 508 such as file I/O, object I/O, and socket I/O streams. Converters 506 have the capability to extract the data portion from stream 502 to support streams 508 at any protocol layer.

According to arrangement 500, any number of protocol-specific streams 508 may be connected to stream 502. In addition, the headers of messages in stream 502 may contain destination-specific encryption/authentication information. For example, stream 502 may contain an encryption/authentication value A, while a recipient of one of the protocol-specific streams 508 holds a key value X, making the decryption of stream 502 a function of A and X ( $\text{key}=\text{f}(A, X)$ ). Likewise, similar keys may be developed for the other protocol-specific streams 508.

## HARDWARE OVERVIEW

Figure 6 is a block diagram which illustrates a computer system 600 upon which an embodiment of the invention may be implemented. Computer system 600 includes a bus 602 or other communication mechanism for communicating information, and a processor 604 coupled with bus 602 for processing information. Computer system 600 also includes a main memory 606, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 602 for storing information and instructions to be executed by processor 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 604. Computer system 600 also includes a read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and instructions for processor 604. A storage device 610, such as a magnetic disk or optical disk, is also provided and coupled to bus 602 for storing information and instructions.

Computer system 600 may also be coupled via bus 602 to a display 612, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 614, including alphanumeric and other keys, is also provided and coupled to bus 602 for communicating information and command selections to processor 604. Another type of user input device is cursor control 616, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 604 and for controlling cursor movement on display 612. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allows the device to specify positions in a plane.

The invention is related to the use of computer system 600 to provide layer-independent secure network communication. According to one embodiment of the invention, layer-independent secure network communication is provided by computer system 600 in response to

processor 604 executing sequences of instructions contained in main memory 606. Such instructions may be read into main memory 606 from another computer-readable medium, such as storage device 610. However, the computer-readable medium is not limited to devices such as storage device 610. For example, the computer-readable medium may include a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, or any other medium from which a computer can read. Execution of the sequences of instructions contained in main memory 606 causes processor 604 to perform the process steps previously described. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

Computer 600 also includes a communication interface 618 coupled to bus 602. Communication interface 608 provides a two-way data communication coupling to a network link 620 to a local network 622. For example, if communication interface 618 is an integrated services digital network (ISDN) card or a modem, communication interface 618 provides a data communication connection to the corresponding type of telephone line. If communication interface 618 is a local area network (LAN) card, communication interface 618 provides a data communication connection to a compatible LAN. Wireless links are also possible. In any such implementation, communication interface 618 sends and receives electrical, electromagnetic or optical signals which carry digital data streams representing various types of information.

Network link 620 typically provides data communication through one or more networks to other data devices. For example, network link 620 may provide a connection through local network 622 to a host computer 624 or to data equipment operated by an Internet Service Provider (ISP) 626. ISP 626 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 628. Local network 622 and Internet 628 both use electrical, electromagnetic or optical signals which carry digital data streams. The signals through the various networks and the signals on network link 620 and through communication interface 618, which carry the digital data to and from computer 600 are exemplary forms of carrier waves transporting the information.

Computer 600 can send messages and receive data, including program code, through the network(s), network link 620 and communication interface 618. In the Internet example, a server 630 might transmit a requested code for an application program through Internet 628, ISP 626, local network 622 and communication inter-

face 618. In accord with the invention, one such downloaded application provides for the synchronization of threads using selective object locking as described herein.

The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution. In this manner, computer 600 may obtain application code in the form of a carrier wave.

Although the invention has been described in the context of connection-based communication architectures, the invention is also applicable to sessionless datagram or packet based communication architectures.

The invention provides several advantages over prior approaches for implementing secure network communications. Most importantly, security is implemented using streams which are layer independent. This allows an encrypted stream to be decrypted at any layer without requiring the use of layer specific calls to perform the decryption, which provides greater flexibility than prior approaches. For example, an encrypted stream transmitted by a sending node may be decrypted by a firewall connection at the network (packet) layer having knowledge of the encryption approach negotiated during system setup. Moreover, this approach does not affect existing encryption being carried out at various layers. The approach of the invention avoids the setup negotiation which can significantly degrade communication performance in certain situations.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

### Claims

1. A method for providing communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol, the method comprising the steps of:

- a) establishing a communication channel between the first network node and the second network node;
- b) establishing a first stream between the first process and the communication channel;
- c) establishing a second stream between the second process and the communication channel;

5 d) encrypting data to be transmitted between the first and second processes, the encrypting of the data being independent of the at least one communication protocol supported by the first network node;

10 e) writing the encrypted data to the first stream; f) causing the encrypted data to be transmitted from the first network node to the second network node;

15 g) reading the encrypted data from the second stream; and

20 h) decrypting the encrypted data to obtain decrypted data which is identical to the data on the first network node before the data was encrypted.

2. The method of Claim 1, further including the steps of

25 a) performing a communication protocol-specific encryption of the data on the first network node, and

30 b) performing a communication protocol-specific decryption of the data on the second network node.

3. The method of Claim 1, wherein the communication channel is a Java secure channel,

35 wherein the first stream is a first Java stream, wherein the second stream is a second Java stream,

40 wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

45 wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

50 wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

55 wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and

60 wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

4. The method of Claim 1, wherein the communication channel is a Java secure channel, wherein the first

stream is a Java stream,

wherein the second stream is a Java stream, wherein the method further comprises the step of connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol.

5

7. The computer-readable medium of Claim 5, wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

10

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

15

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

20

wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and

25

wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

30

8. The computer-readable medium of Claim 5, wherein the communication channel is a Java secure channel,

35

wherein the first stream is a Java stream, wherein the second stream is a Java stream, wherein the computer-readable medium further includes instructions for connecting the Java secure channel to a third Java stream, and

40

wherein the third Java stream provides for the transmission of data according to a specific communication protocol.

45

9. A communication network providing communication protocol-independent secure communication between a first network node and a second network node, wherein the first network node and the second network node each support at least one common communication protocol, wherein the first network node is communicatively coupled to the second network node by a communication channel, the communication network comprising:

50

a) a first process executing on the first network node, wherein the first process provides for the communication protocol-independent encryption of data;

55

b) a first stream which provides for the transfer

60

65

70

75

80

85

90

95

of encrypted data between the first process and the communication channel;

c) a second process executing on the second network node; and

d) a second stream which provides for the transfer of encrypted data between the communication channel and the second process, wherein the second process also provides for the decryption of data which has been encrypted by the first process.

10. The communication network of Claim 9, wherein the second process further includes the capability to decrypt data based upon any communication protocol supported by the second network node.

11. The communication network of Claim 9, wherein the communication channel is a Java secure channel, the first stream is a Java stream and the second stream is a Java stream.

12. The communication network of Claim 11, further comprising a third Java stream connected to the Java secure channel, the third Java stream providing for the transmission of data according to a specific communication protocol.

13. A computer data signal embodied in a carrier wave and representing sequences of instruction which, when executed by one or more processors, provide communication protocol-independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol by performing the steps of:

- a) establishing a communication channel between the first network node and the second network node;
- b) establishing a first stream between the first process and the communication channel;
- c) establishing a second stream between the second process and the communication channel;
- d) encrypting data to be transmitted between the first and second processes, the encrypting of the data being independent of the communication protocols supported by the first network node;
- e) writing the encrypted data to the first stream;
- f) causing the encrypted data to be transmitted from the first network node to the second network node;
- g) reading the encrypted data from the second stream; and
- h) decrypting the encrypted data to obtain de-

14. The computer data signal of Claim 13, wherein the computer sequence of instructions further includes instructions for performing the steps of

- a) performing a communication protocol-specific encryption of the data on the first network node, and
- b) performing a communication protocol-specific decryption of the data on the second network node.

15. The computer data signal of Claim 13, wherein the first stream is a first Java stream,

20 wherein the second stream is a second Java stream,  
wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,  
wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,  
wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel,

25 wherein the step of writing the encrypted data to the first stream further comprises the step of writing the encrypted data to the first Java stream, and  
wherein the step of reading the encrypted data from the second stream further comprises the step of reading the encrypted data from the second Java stream.

30 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 9999 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10099 10100 10101 10102 10103 10104 10105 10106 10107 10108 10109 10110 10111 10112 10113 10114 10115 10116 10117 10118 10119 10120 10121 10122 10123 10124 10125 10126 10127 10128 10129 10130 10131 10132 10133 10134 10135 10136 10137 10138 10139 10140 10141 10142 10143 10144 10145 10

dependent security for data transmitted by a process executing on a network node, the method comprising the steps of:

- a) establishing a stream between the process and a communication channel;
- b) encrypting data to be transmitted by the process, the encrypting of the data being independent of a communication protocol supported by the network node;
- c) writing the encrypted data to the stream; and
- d) causing the encrypted data to be transmitted from the network node to the communication channel.

15

18. The method of Claim 17, wherein the communication channel is a Java secure channel,

wherein the stream is a first Java stream, 20  
wherein the step of establishing a stream between the process and the communication channel further comprises the step of establishing a Java stream between the process and the Java secure channel, and  
wherein the step of writing the encrypted data to the stream further comprises the step of writing the encrypted data to the Java stream.

25

19. The method of Claim 17, wherein the communication channel is a Java secure channel, wherein the stream is a Java stream, 30

wherein the method further comprises the step of connecting the Java secure channel to a second Java stream, and  
wherein the second Java stream provides for the transmission of data according to a specific communication protocol.

35

40

45

50

55

10

FIG. 1

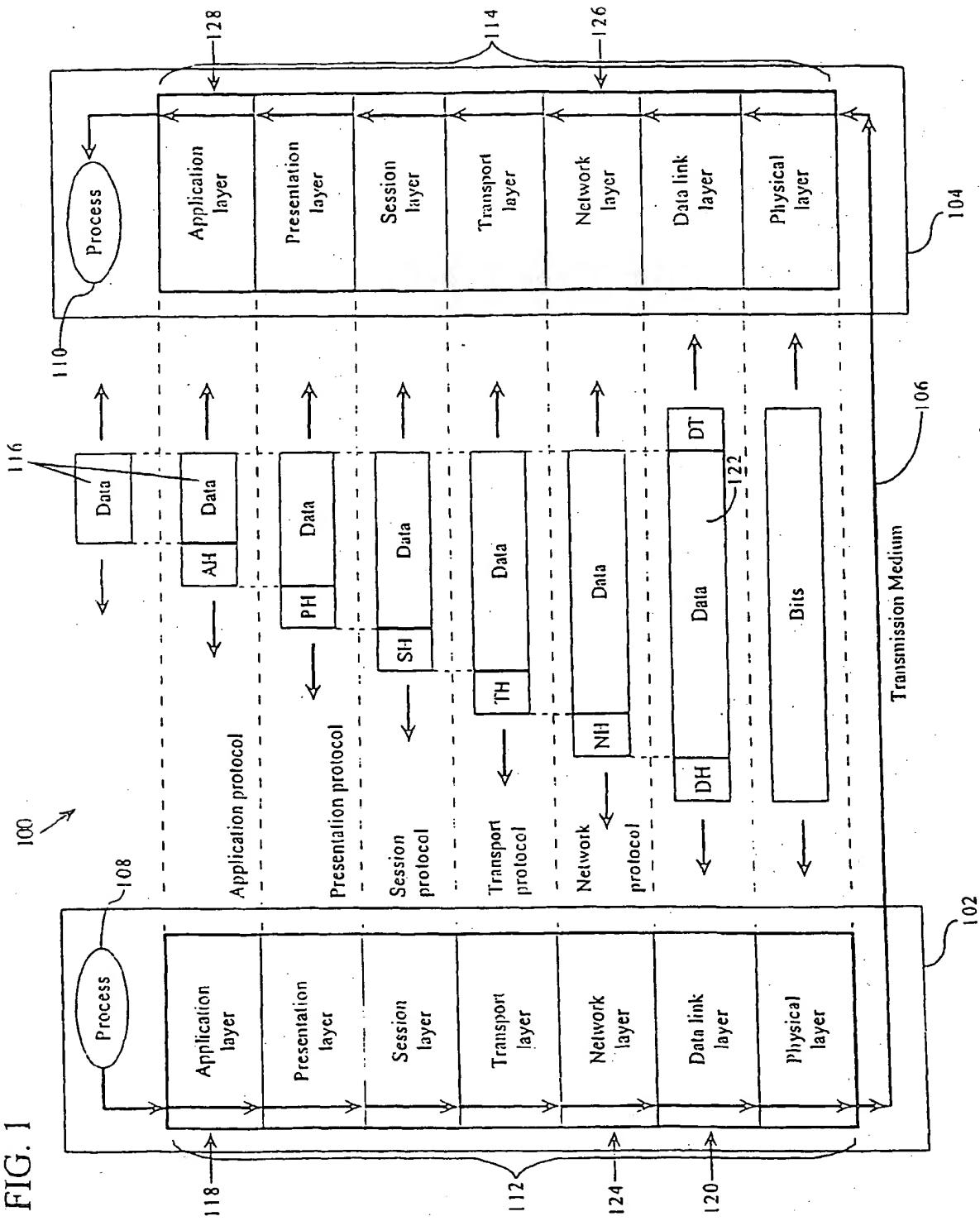


FIG. 2

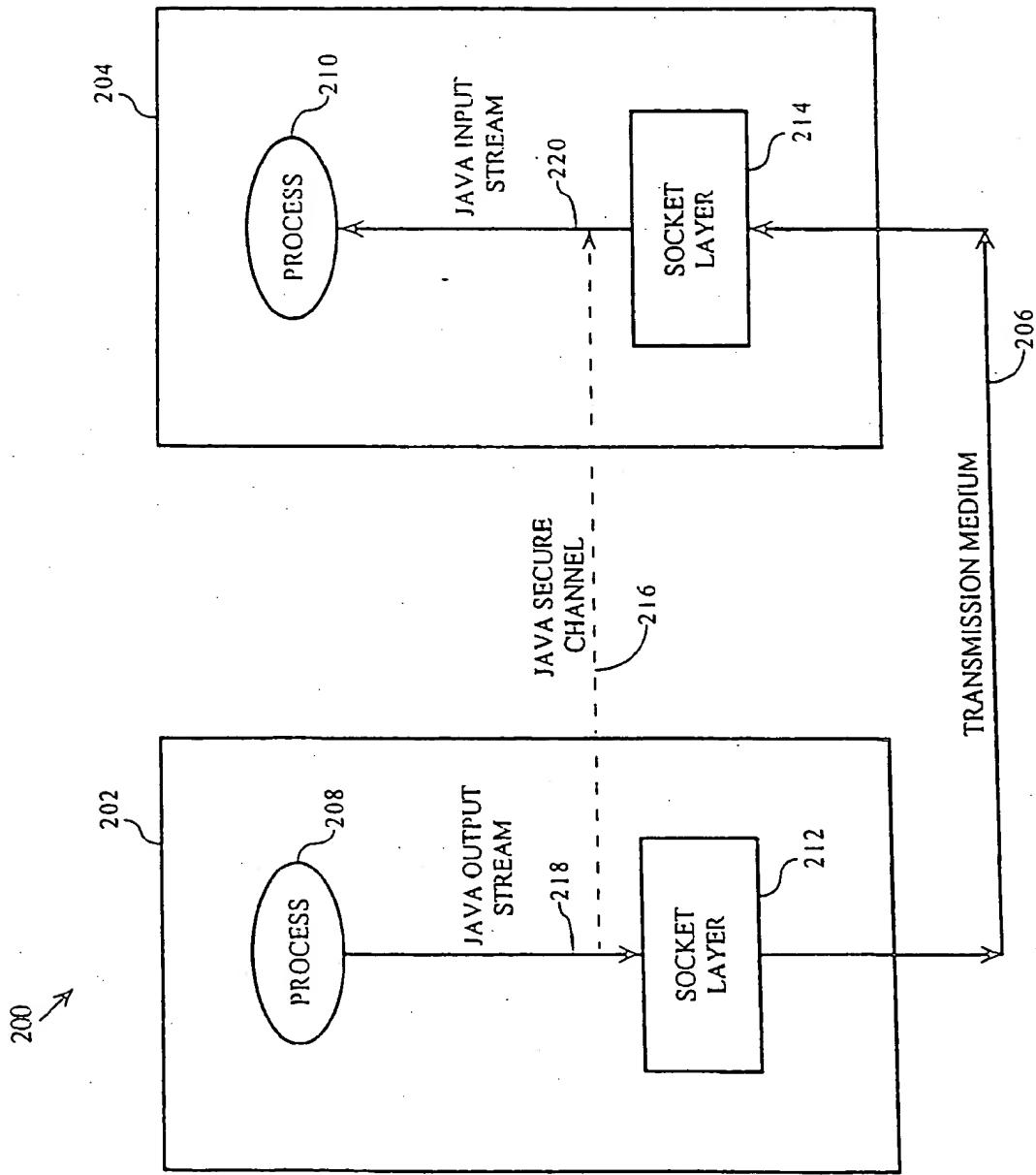


FIG. 3

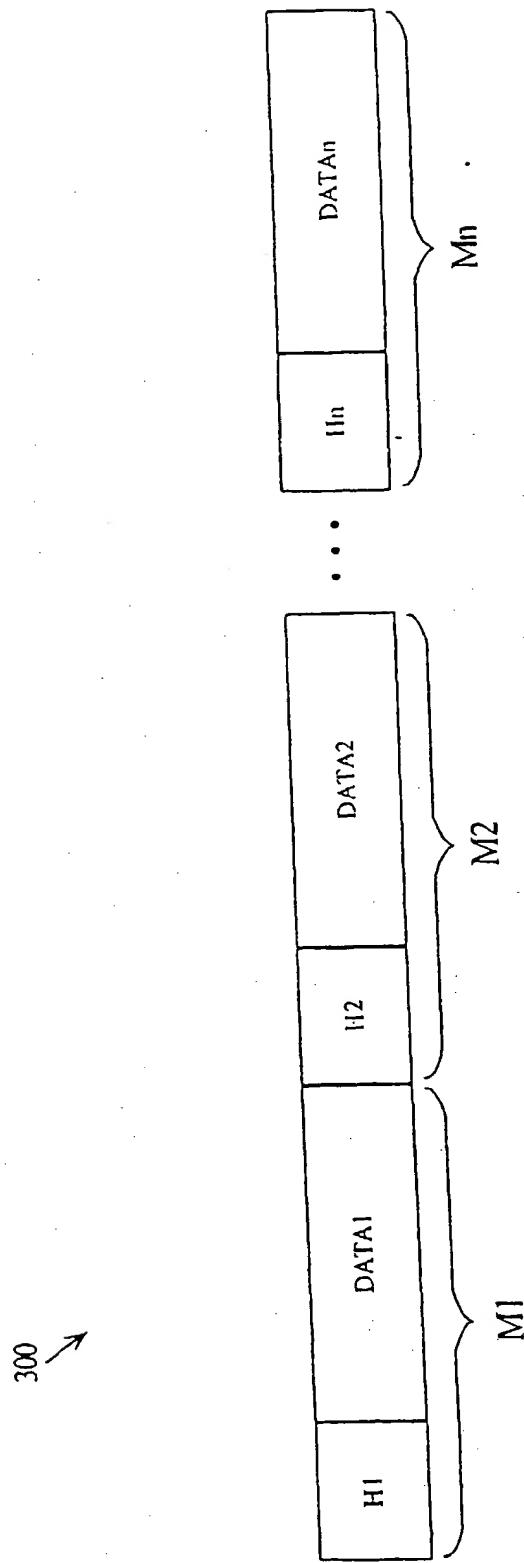


FIG. 4

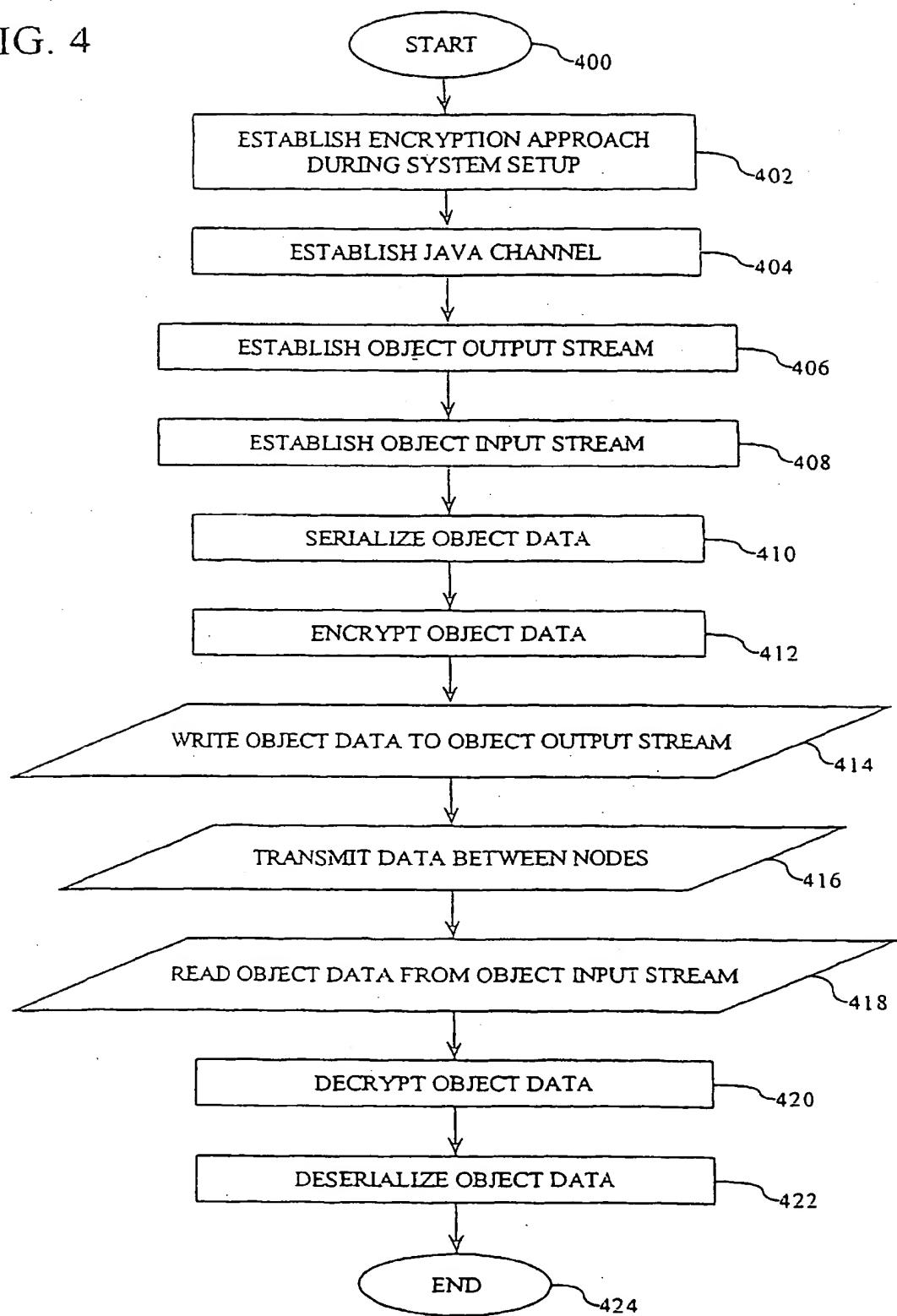


FIG. 5

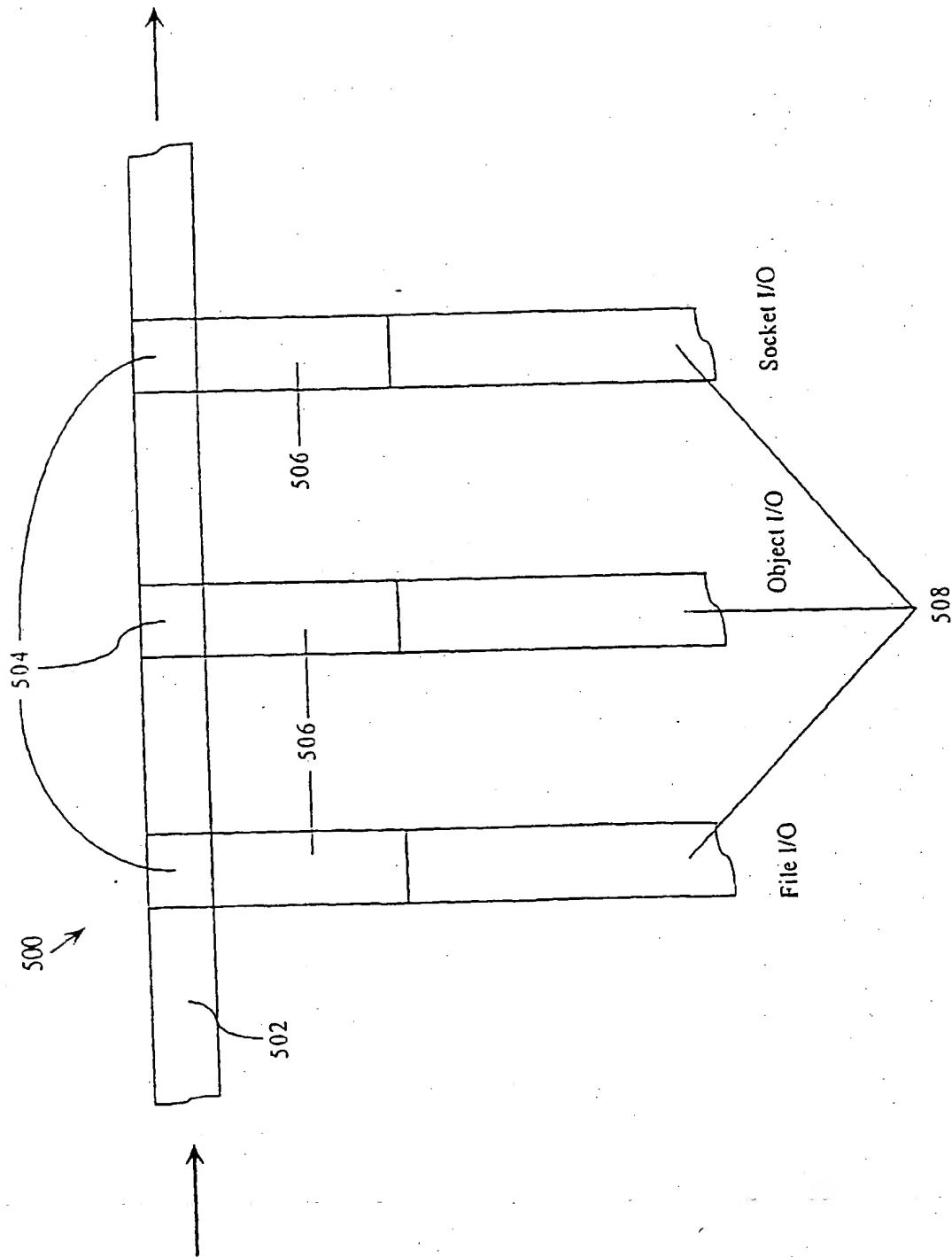
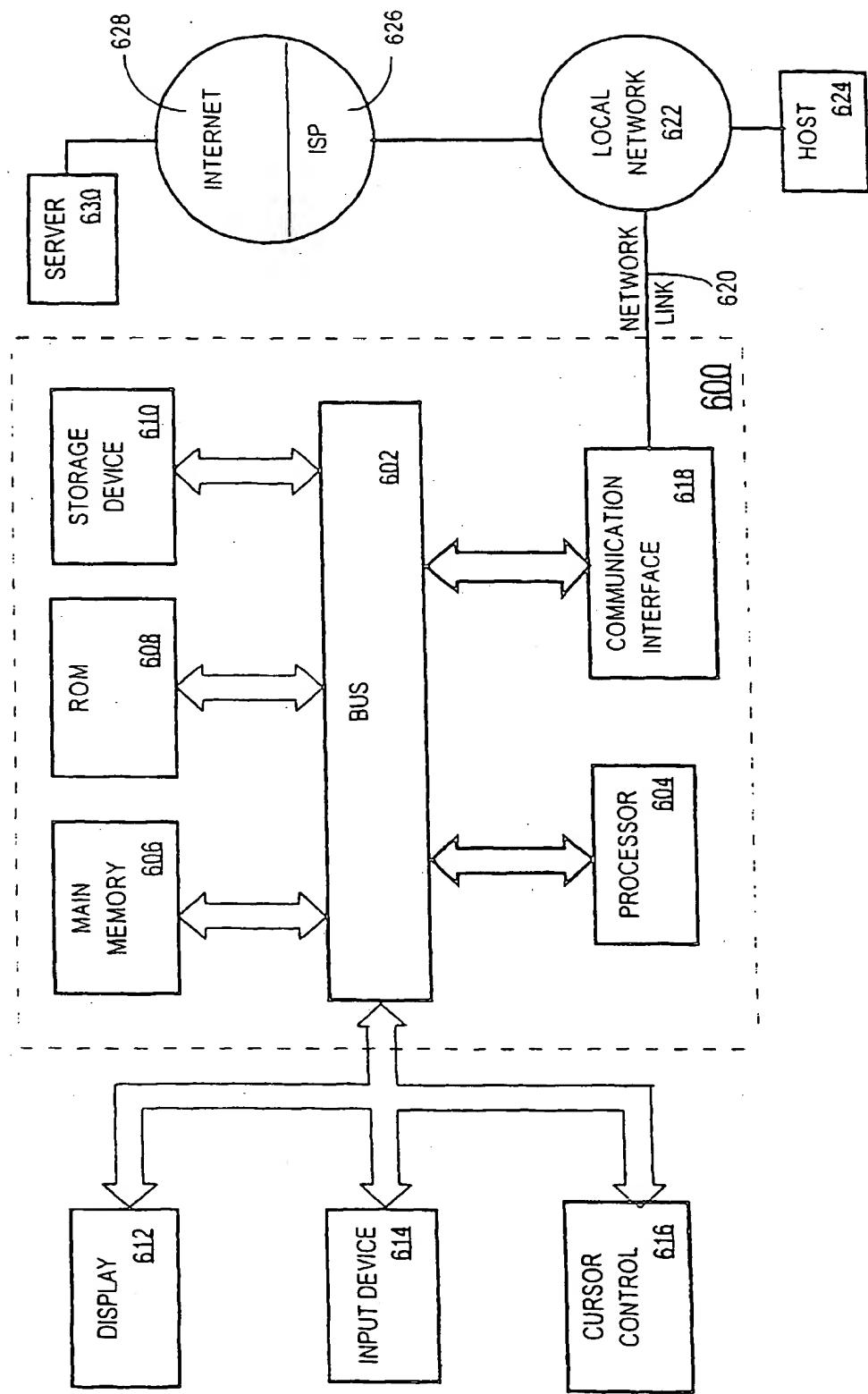
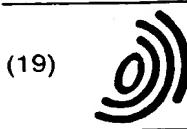


FIG. 6





(12)

## EUROPEAN PATENT APPLICATION

(88) Date of publication A3:  
28.07.1999 Bulletin 1999/30

(51) Int Cl. 6: H04L 29/06

(43) Date of publication A2:  
30.12.1998 Bulletin 1998/53

(21) Application number: 98304869.5

(22) Date of filing: 19.06.1998

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE

Designated Extension States:  
AL LT LV MK RO SI

(30) Priority: 26.06.1997 US 883636

(71) Applicant: SUN MICROSYSTEMS, INC.  
Mountain View, California 94043-1100 (US)

(72) Inventor: Gong, Li  
Menlo Park, California 94025 (US)

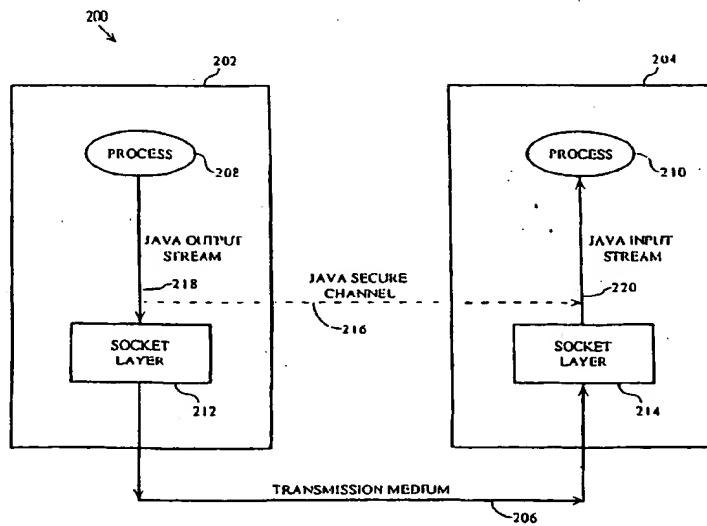
(74) Representative: Read, Matthew Charles et al  
Venner Shipley & Co.  
20 Little Britain  
London EC1A 7DH (GB)

### (54) Layer-independent security for communication channels

(57) A method and apparatus for providing layer-independent secure network communication is provided. According to an embodiment of the invention, a transmission medium (206) is provided between a first network node (200) and a second network node (204). Both the first network node and the second network node support at least one common communication protocol. A Java output stream (218) is established between a first process (208) executing on the first network node

and the transmission medium. Also, a Java input stream (220) is established between a second process (210) executing on the second multilayered node and the transmission medium. Data to be transmitted from the first process to the second process is encrypted by the first process and written to the Java output stream. The data is transmitted to the second network node. Then the data is read from the Java input stream by the second process and decrypted.

FIG. 2





European Patent  
Office

## EUROPEAN SEARCH REPORT

Application Number  
EP 98 30 4869

DOCUMENTS CONSIDERED TO BE RELEVANT									
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)						
X	EP 0 778 512 A (SUN MICROSYSTEMS INC) 11 June 1997  * abstract * * page 2, column 2, line 15 - line 44 * * page 3, column 4, line 17 - page 4, column 6, line 39 * * page 6, column 10, line 24 - line 39 * ---	1,3-9, 11,13, 15,17,18	H04L29/06						
Y	WO 95 25396 A (TRANSACTION TECHNOLOGY INC) 21 September 1995  * abstract * * page 13, line 20 - page 18, line 10 * * page 35, line 16 - page 36, line 6 * * page 44, line 12 - page 45, line 25 * * page 47, line 22 - page 48, line 10 * * page 58, line 3 - line 21 * * figure 16 * ---	2,10,12, 14,19							
A	EP 0 776 112 A (SUN MICROSYSTEMS INC) 28 May 1997  * abstract * * page 2, column 2, line 5 - line 53 * * page 3, column 3, line 45 - column 4, line 8 * * page 4, column 6, line 41 - page 5, column 7, line 3 * * claims 1,2 * ---	1,5,9, 13,17	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04L						
A	ZUQUETE A ET AL: "TRANSPARENT AUTENTICATION AND CONFIDENTIALITY FOR STREAM SOCKETS" IEEE MICRO, vol. 16, no. 3, 1 June 1996, pages 34-41, XP000594076 * the whole document * ---	1,5,9, 13,17							
<p>The present search report has been drawn up for all claims</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">Place of search</td> <td style="width: 33%;">Date of compilation of the search</td> <td style="width: 34%;">Examiner</td> </tr> <tr> <td>THE HAGUE</td> <td>21 May 1999</td> <td>Poggio, F</td> </tr> </table>				Place of search	Date of compilation of the search	Examiner	THE HAGUE	21 May 1999	Poggio, F
Place of search	Date of compilation of the search	Examiner							
THE HAGUE	21 May 1999	Poggio, F							
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons R : member of the same patent family, corresponding document							
X : particularly relevant & taken alone Y : particularly relevant & combined with another document of the same category A : technological background C : non-written disclosure P : intermediate document									

ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.

EP 98 30 4869

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.  
 The members are as contained in the European Patent Office EDP file on  
 The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-05-1999

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP 0778512 A	11-06-1997	US	5708709 A	13-01-1998
		JP	9288575 A	04-11-1997
WO 9525396 A	21-09-1995	US	5572572 A	05-11-1996
		AU	2123495 A	03-10-1995
		CA	2185662 A	21-09-1995
		CN	1146262 A	26-03-1997
		EP	0750817 A	02-01-1997
EP 0776112 A	28-05-1997	US	5680461 A	21-10-1997
		AU	6793096 A	01-05-1997
		CA	2187970 A	27-04-1997
		JP	9270788 A	14-10-1997
		US	5850449 A	15-12-1998

**This Page Blank (uspto)**